

# Index

## A

- Abandon **method**, **session state**, 422–423, 436, 447
- access control lists**. *See* **ACLs (access control lists)**
- account lockouts**, 617–621
- ACE (ASP.NET and Application Consulting & Engineering) teams**, 855–856
- ACLs (access control lists)**
  - FileAuthorizationModule checking, 123–124
  - IUSR account, 80
  - managing Active Directory default, 667
  - reading local configuration, 247–249
  - SSE user instances, 578–579
  - writing local configuration, 249–251
- AcquireRequestState **event**, 427–429, 435
- Active Directory Lightweight Directory Services**. *See* **ADLDS (Active Directory Lightweight Directory Services)**
- Active Directory Lightweight Directory Services Setup Wizard**, 676–680
- Active Directory Schema editor**, 667–668
- ActiveDirectoryMembershipProvider, 639–690
  - ActiveDirectoryMembershipUser, 654–657
  - configuration settings, 649–650
  - configuring ASP.NET membership, 792
  - connection settings, 642–645
  - containers, nesting, 660–662
  - containers, securing, 662–667
  - defined, 525
  - error-handling approaches, 550–551
  - of MembershipProvider class, 537
  - overview of, 639
  - in partial trust, 684–689
  - primary key, 552–553
  - schema mappings, 645–648
  - search settings, 648–649
  - self-service password reset, 667–675
  - summary review, 689–690
  - supported architectures, 640–642
  - unique functionality of, 651–654
  - UPNs and SAMAccountName, 659–660
  - using ADLDS. *See* **ADLDS (Active Directory Lightweight Directory Services)**
  - UTC time, 536
  - working with, 657–659
- ActiveDirectoryMembershipUser, 654–657
- AddOnPostAuthenticateRequestAsync, 109
- Add-Remove-Clear (ARC) collections**, 230–232
- AddUsersToRole **method**, 694, 766
- AddUsersToRoles **method**
  - RolePrincipal/Roles classes, 694
  - RoleProvider, 725
  - SqlRoleProvider, 738
- AddUserToRole **method**, 694
- AddUserToRoles **method**, 694
- ADLDS (Active Directory Lightweight Directory Services)**, 675–684
  - connection settings, 642–645
  - enabling Role Manager, 697
  - installing with application partition, 676–682
  - overview of, 675–676
  - supported directory architectures, 640–642
  - using application partition, 682–684
- administration, IIS 7.0 improvements**, 6–9
- Administration.config **file**, 3
- ADODB, sandboxed access to**, 208–209
- adsiedit MMC tool**, 679
- Advanced Encryption Standard (AES)**, 299–301
- AES (Advanced Encryption Standard)**, 299–301
- AJAX 3.5, 791–821**
  - ASP.NET membership, 792–794
  - ASP.NET role management, 794–796
  - authentication service, 804–814
  - AuthenticationServiceManager class, 801–803
  - enabling application services, 801–803
  - enabling ASP.NET applications, 796–801
  - overview of, 791
  - role service, 815–820
  - RoleServiceManager class, 803–804
  - summary review, 820–821
- AJAX-enabled application threats**, 871–877
  - amplified cross-site scripting, 875–877
  - information leakage, 871–874
  - JSON hijacking, 874–875
  - overview of, 826
- allowOverride **attribute**, <location /> **element**, 226
  - configuring trust levels, 150–151
  - finding trust policy file, 155
  - overview of, 226
  - processRequestInApplicationTrust, 214, 221

# AllowPartiallyTrustedCallersAttribute

---

AllowPartiallyTrustedCallersAttribute  
**(APTCA), 198–204**

AllowRemoteConnection, **OOB state server, 447**  
**anonymous access**

- authentication best practices, 827
- EndRequest, RoleManagerModule, 711
- IIS 7.0 security improvements, 12
- PostAuthenticateRequest, RoleManagerModule, 709
- as security option, 82–83

AnonymousAuthenticationModule

- <authentication>, 38–39
- ASP.NET Integrated mode advantages, 33
- changing default identity of application pools, 93
- hooking PostAuthenticateRequest, 120–122
- impersonation token for, 93
- security configuration, 83–84
- understanding, 85–87

**Anti-Cross Site Scripting Library, 855–857**

**API, IIS 7.0, 3**

appcmd.exe, **8, 12**

app.config **file, 554–555**

\$AppDir\$ **string replacement token, 156, 160–161**

\$AppDirUrl\$ **string replacement token, 156**

**application domains**

- initiating per-request security, 82
- working with trust levels, 163

**application identity, authentication best practices, 827**

**application partitions, ADLDS, 676–684**

**Application Pool Identity account, 86**

**application pools**

- changing default identity associated with, 92
- Classic mode, 18–19
- Integrated mode, 18
- overview of, 17–18
- removing and editing, 7

Application\_Error **event, global error handling, 864**

applicationHost.config **file**

- <system.webServer />, 34–37
- administration of, 6–7
- authenticating classic ASP, 395
- configuring AnonymousAuthenticationModule, 85–87
- IIS 7.0 configuration based on, 3, 233–234
- IIS 7.0 feature delegation, 238–243
- managing output caching module, 236
- Windows Process Activation Service and, 21

ApplicationId

- aspnet\_Membership table, 574
- aspnet\_Roles table, 736
- common users table, 564
- resolving application name to, 569–570

applicationName **attribute**

- ActiveDirectoryMembershipProvider, 649
- as Membership feature primary key, 552
- MembershipProvider, 541
- storing application name, 562–563
- supporting dynamic applications, 626–632
- using Membership outside of ASP.NET, 554–555
- using Role Manager with membership, 786–788

ApplicationName **property**

- ActiveDirectoryMembershipProvider, 654
- AuthorizationStoreRoleProvider, 767
- RoleProvider, 724
- Roles class, 692
- SqlRoleProvider, 757–758

**APTCA (AllowPartiallyTrustedCallers Attribute), 198–204**

**ARC (Add-Remove-Clear) collections, 230–232**

**ASP.NET**

- AJAX 3.5. See AJAX 3.5
- application services, 801–803
- authorizing classic ASP. See authorization, classic ASP with ASP.NET
- enabling applications with AJAX, 796–801
- encryption, 299–303
- IIS 7.0 configuration vs., 233–235
- IIS request pipeline and. See IIS (Internet Information Services) 7.0, Integrated mode
- membership, 792–794
- permission set. See permission sets
- role management, 794–796
- sharing tickets between versions, 324–325
- using Membership feature outside of, 553–554

**ASP.NET and Application Consulting & Engineering (ACE) teams, 855–856**

**ASP.NET security, integrating with classic ASP, 373–415**

- authentication, 389–395
- authorization, 396–414
- DefaultHttpHandler, 383–384
- IIS 5 ISAPI extensions, 374–375
- IIS 7 wildcard mappings, 375–383
- overview of, 373–374
- servicing classic ASP in IIS integrated mode, 387–388
- using DefaultHttpHandler, 384–387

**ASP.NET security, web application best practices, 823–878**

- AJAX-enabled application threats, 871–877
- authenticate users, 827–828
- authorize users, 828
- cross-site request forgery, 857–861
- cross-site scripting, 853–857
- database access, encrypt data, 845–849
- database access, overview, 841
- database access, SQL Server, 843–845
- database access, Windows authentication, 842–843
- DOS threats, 865–871

- exception handling, 861–865
- minimizing privileges, 829
- overview of, 823–824
- secure cookies, 838–841
- secure data transmission, 871
- SQL injection attacks, 849–853
- summary review, 877–878
- threats, 824–826
- validate user input, 829–838
- aspnet\_Applications **table**
  - aspnet\_Users and, 564
  - linking custom features to user records, 570–572
  - querying with views, 568
  - storing application names, 563
- aspnet\_CheckSchemaVersion, **568**
- aspnet\_compiler, **455**
- aspnet\_Membership **table**, **573–576**
- aspnet\_regiis **tool**
  - command-line options, 273–274
  - configuring keyContainerName, 270
  - configuring useMachineContainer, 272
  - protected configuration, 259–260
  - protected configuration providers, 264
  - remote editing permissions, 252–253
  - synchronizing key containers, 272–273
  - useMachineProtection and, 267
- aspnet\_regsql.exe **tool**, **446**
- aspnet\_Roles, SqlRoleProvider, **736–737**
- aspnet\_Roles\_BasicAccess, SqlRoleProvider, **745**
- aspnet\_Roles\_FullAccess, SqlRoleProvider, **745**
- aspnet\_Roles\_ReportingAccess, SqlRoleProvider, **745**
- aspnet\_SchemaVersions **table**, **566–568**
- aspnet\_Users **table**
  - linking custom features to user records, 570–572
  - overview of, 563–566
  - password history, 602–604
  - querying with views, 568
  - vw\_aspnet\_MembershipUsers view and, 576
- aspnet\_UsersInRoles **table**, SqlRoleProvider, **737–739**
- aspnetdb, **SSE connection string**, **584**
- AspNetHostingPermission **class**, **182–187**
  - AuthorizationStoreRoleProvider, 783–785
  - outside ASP.NET, 185–186
  - overview of, 182
  - restricted by trust level, 184–185
  - SqlRoleProvider, 741–742, 745
  - trust level intent, 183–184
  - using in code, 186–187
- aspnet-regsql **tool**, **584**
- ASPState**, **445–446**
  - .aspx **login file**, **394**
  - assemblies, signing precompiled**, **455–457**
  - assemblies, strongly named**
    - APTCA and, 200–204
    - sandboxing with, 204–208
  - <assemblies /> **configuration section, AJAX**, **798**
  - Assert **method, sandboxing**, **208, 210–213**
  - Assertion permission**, **192–193**
  - asynchronous execution**, **137–143**
    - overview of, 137–138
    - PreRender processing, 138–141
    - using PageAsyncTask, 141–143
  - asynchronous page tasks**, **141–143**
  - asynchronous pipeline events**, **100–110**
  - attributeMapUserName,
    - ActiveDirectoryMembershipProvider, **660**
  - attributes, locking**, **227–229**
  - AUTH\_TYPE, WindowsAuthenticationModule, **112–113**
  - Authenticate **event**, **113–114, 115–116**
  - Authenticated user, ASP.NET impersonation**, **97**
  - AuthenticateRequest **event**, **110–117**
    - enforcing single logins, 363–365
    - forms authentication recap, 288
    - FormsAuthenticationModule, 115–117
    - forwarding request to EndRequest event, 143–144
    - overview of, 110–111
    - RoleManagerModule, 709–710
    - WindowsAuthenticationModule, 111–115
  - authentication**
    - AnonymousAuthenticationModule, 85–87
    - ASP.NET membership, 793
    - AuthenticateRequest event, 110–117
    - best practices, 827–828
    - classic mode, 31–32, 389–395
    - cross-application sharing of ticket, 333–334
    - DefaultAuthentication and Thread.
      - CurrentPrincipal, 117–120
    - forms. See forms authentication
    - impersonation tokens, 93
    - Integrated mode, 32–33
    - Role Manager, 696–697
    - RoleManagerModule, 709
    - security, 81–84
    - Windows. See Windows authentication
  - authentication, AJAX 3.5**, **804–814**
    - check if user is authenticated, 812
    - custom authentication service, 812–814
    - login user, 808–811
    - logout user, 812
    - overview of, 804
    - Sys.Services.AuthenticationService class, 805–807
    - System.Web.ApplicationServices.
      - AuthenticationService class, 805
  - <authentication> **configuration section**

## <authentication> configuration section (*continued*)

---

### <authentication> **configuration section** (*continued*)

- authenticating classic ASP, 394–395
- configuring ASP.NET membership, 793
- enabling ASP.NET application services, 801
- HTTP request processing, 85
- overview of, 38–40
- AuthenticationService **class**, 804, 815
- AuthenticationServiceManager **class**, 801–803
- authorization**
  - ASP.NET configuration section, 40–42
  - authentication vs., 828
  - configuring role management, 795
  - FileAuthorizationModule, 123–124
  - IIS 7.0 security improvements, 11
  - RoleProvider methods, 724–725
  - with roles in data layer, 755–757
  - security best practices, 828
  - UrlAuthorizationModule. See UrlAuthorizationModule
- authorization, classic ASP with ASP.NET**, 396–410
  - full code listing of hash Helper, 406–410
  - overview of, 396
  - passing sensitive data to classic ASP, 398–406
  - passing user roles to classic ASP, 397–398
- Authorization Manager**. See AzMan (Authorization Manager)
- AuthorizationRules icon**, 130
- AuthorizationSection **class**,
  - UrlAuthorizationModule, 125, 127
- AuthorizationStoreRoleProvider, 763–789
  - design, 763–766
  - functionality of, 766–768
  - overview of, 763
  - in partial trust, 783–785
  - role management, 795
  - role nesting, 724
  - summary review, 789
  - using directory-based policy store, 771–779
  - using file-based policy store, 768–771
  - using Membership with Role Manager, 786–789
  - using SQL Server database-based policy store, 780–783
- AuthorizeRequest **event**, 122–135
  - EndRequest event, 143–144
  - FileAuthorizationModule, 123–124
  - forms authentication, 288
  - managed UrlAuthorizationModule, 124–129
  - native UrlAuthorizationModule, 129–135
  - operating system thread identity and, 100
  - overview of, 122–123
  - PostAuthorizeRequest, 135
- AutoDetect, cookieless **attribute**
  - defined, 309
  - issuing cookieless session IDs, 424
  - overview of, 310–313

### **auto-generated keys, and encryption**, 300–301

- automatic unlocking**, 621–626
  - autoUnlockTimeout, SqlMembershipProvider, 622–626
- AzMan (Authorization Manager)**, 783–785
  - AuthorizationStoreRoleProvider using, 724, 764–768
  - enabling, 697
  - overview of, 764
  - in partial trust, 783–785
  - using directory-based policy store, 771–779
  - using file-based policy store, 768–771
  - using Membership with Role Manager, 785–789
  - using Microsoft SQL Server database-based policy store, 780–783

## B

### **backwards compatibility**, 214–221

### **BasicAccess role, database**, 585–586, 745

- BasicAuthenticationModule
  - <authentication>, 38
  - <modules />, 35
  - impersonation token, 93–95
- Begin **event handler**, 104, 138–141
- BeginInvoke **method, asynchronous pipeline events**, 104–105
- BeginProcessRequest, DefaultHttpHandler, 385–386

### **bin directory**, 202–204

### **blocking requests, HTTP**, 135–137

### **Boolean values**, MembershipProvider, 551

### **browsers**

- AutoDetect, 310–313
- cookieless tickets/other URLs in pages, 317–319
- replay attacks, 315–316
- UseDeviceProfile, 313–315

## C

### **C#**

- ActiveDirectoryMembershipProvider, 658–659, 684
- ActiveDirectoryMembershipUser, 656
- AJAX, 813, 819
- AllowPartiallyTrustedCallersAttribute, 199, 201, 203
- ASP.NET membership, 793
- ASP.NET role management, 795
- AspNetHostingPermission class, 185–187
- asynchronous page tasks, 142
- asynchronous pipeline events, 101–108
- asynchronous PreRender processing, 138–141
- authenticating classic ASP with ASP.NET, 390
- authorizing classic ASP with ASP.NET, 397

- building provider-based feature, 495–498, 500–502, 505–511, 515
- clock resets and, 292–293
- container nesting, 661
- cookie settings, 304–305
- cookie timeout, 290
- cookie-based SSO-lite, 347, 349, 352–353
- cookieless cross-application behavior, 335–338
- cookieless forms authentication for classic ASP, 391
- cross-site request forgery, 858–860
- cross-site scripting protection, 855–857
- cross-site scripting threat in AJAX, 876–877
- custom Hash algorithms, 557–559
- custom password encryption, 594–597
- custom passwords, 591–594
- customizing configuration providers, 279, 281–285
- customizing `OdbcPermission`, 174–175
- customizing `OleDbPermission`, 171–173
- directory-based policy store, 776–777
- DoS attack protection, 865–871
- encrypting data, 847–848
- enforcing logouts, 369–370
- enforcing single logins, 360–361, 363–364, 366
- Factory Method pattern, 477–479
- filtering data before encoding, 832–834
- fraudulent postbacks, 459
- generating keys, 301–303
- global error handling, 865
- hash `Helper`, 407–408
- JSON hijacking threat, 875
- `LinkDemand` exception behavior, 195–197
- LINQ trust levels, 180
- local configuration, 250
- locating identity for requests, 87–90
- managed handlers, 50, 53–54, 56–65, 69–71
- managed modules, 67–68, 73–76
- Membership supported environments, 555
- `MembershipProvider` class, 538–539
- `Microsoft.Web.Administration`, 7
- migrating ASP.NET applications, 43–44, 47
- partial trust, 253
- passing data to ASP from ASP.NET, 392–393
- passing data to classic ASP, 399–406, 412
- permission sets, 165–167
- permissions in policy file, 168–171
- persistent cookies, 289
- `PostAuthenticateRequest`, 708, 710
- preventing SQL injection, 850–853
- `processRequestInApplicationTrust`, 215, 219–220
- protected configuration in partial trust, 275–278
- reading and writing configuration, 244–246
- Role Manager with membership, 786–789
- `RoleManagerModule`, 715–721
- `RolePrincipal` class, 695, 700–704
- `RoleProvider` class, 722
- `Roles.DeleteCookie`, 693
- sandboxed access to ADODB, 208–211
- secure cookies, 839–841
- securing containers, 667
- self-service password resets, 673
- serialization and, 442–444
- session state in IIS 7 Integrated mode, 429–431
- signed tickets, 296–297
- site navigation security, 466
- `SqlMembershipProvider`, automatic unlocking, 622–625
- `SqlMembershipProvider`, dynamic applications, 626, 628–629
- `SqlMembershipProvider`, password history, 605–617
- `SqlMembershipProvider`, password strength, 599–601
- `SqlRoleProvider`, dynamic applications, 757–758
- `SqlRoleProvider`, limited set of roles, 748–754
- `SqlRoleProvider` in partially trusted non-ASP.NET, 740–741
- `SqlRoleProvider` in Windows authentication, 746–748
- `System.Configuration` classes, 490, 493–494
- `System.Configuration.Provider` classes, 484, 486–487, 489
- `System.Web.Configuration` classes, 489
- tracing system, 13–15
- trust levels, 152–154
- Try/Catch blocks, 863–864
- `UserData` property, 329–332
- validating user input, 831
- verifying data input, 836–837
- `WebPermission`, 177–179
- wildcard mappings, 379, 381
- `WindowsTokenRoleProvider`, 728, 730–732
- CachedListChanged property, RolePrincipal class, 706**
- cacheRefreshInterval attribute, AuthorizationStoreRoleProvider, 767–768**
- cacheRolesInCookie attribute, role cache cookies, 712**
- CacheRolesInCookie property, Roles class, 692**
- CAPI (Windows cryptographic API), 268**
- CAS (Code Access Security), .NET Framework**
  - history of, 147
  - in partially trusted non-ASP.NET applications, 739, 741–742
  - provider security, 741–742
  - in session state mode, 441
  - with signing precompiled assemblies, 455
  - as trust levels. See trust levels
  - using `ConfigurationPermission`, 256

### case-sensitivity

ASP.NET features, 572  
RolePrincipal class, 699  
caspol.exe tool, 742–744

### certificate mapping, 93

ChangePassword method, 543, 651, 673

ChangePassword method,

    SqlMembershipProvider

    customizing password generation, 590  
    password history, 602, 605–606, 612, 615–617  
    password strength, 598–600  
ChangePasswordQuestionAndAnswer method, 548, 651, 673

### character sets, and URL authorization, 129

<CipherValue /> elements, 270

### classic mode. See also ASP.NET security, integrating with classic ASP

    application pool, 18–19, 26  
    defined, 10  
    IIS 7.0 running in, 31

clientSearchTimeout attribute, 648–649

### code evidence, 159

CodeAccessPermission class, 158–159

\$CodeGen\$ string replacement token, 156–157

<CodeGroup /> elements, 160–161

commandTimeout, SqlMembershipProvider, 577

commandTimeout, SqlRoleProvider, 737–739

Comment column, aspnet\_Membership table, 576

Comment property, MembershipUser, 524, 530–533

Comments field, MembershipUser

    AD/AD LDS directory schema mappings, 645, 647  
    encoding, 831–832  
    enforcing logouts, 369–371  
    enforcing single logins, 367–368

COMMIT TRANSACTION, SqlRoleProvider, 738

### common users table, 563–566

compatibilityMode, <machineKey />, 454

### compilation. See pages and compilation

### components, IIS 7.0, 19–21

<configProtectedData /> section, 261, 264

<configSections /> configuration section, AJAX, 797–798

### configuration, IIS 7.0, 4–6

### configuration API, 259

### configuration manager, Windows Process Activation Service, 20–21

### configuration system security, 223–285

    feature delegation, 238–243  
    IIS 7.0 vs. ASP.NET, 233–235  
    with managed modules/handlers, 236  
    native vs. managed, 236–238  
    protected configuration. See protected configuration  
    reading and writing configuration, 244–253  
    using <location /> element, 223–226  
    using lock attributes, 227–233

ConfigurationErrorsException, 204

ConfigurationManager class

    protected configuration providers in partial trust, 276  
    reading configuration, 244–247  
    reading local configuration, 247–249  
    writing local configuration, 250–251

ConfigurationPermission class, 254–257, 276

ConfigureApplicationService method,

    ScriptManager, 803–804

connectionPassword attribute, 663

ConnectionProtection attribute, 643–645, 654

connectionStringName attribute

    ActiveDirectoryMembershipProvider, 643  
    building provider-based feature, 510–515  
    Factory Method pattern, 479–480  
    redirecting configuration with custom provider, 280–283

    SqlMembershipProvider, 576, 607

    SqlRoleProvider, 737

<connectionStrings /> configuration element

    ActiveDirectoryMembershipProvider, 642–645, 657–659

    building provider-based feature, 510, 512–515

    changing SSE connection string, 583–584

    Factory Method pattern, 479

    SqlMembershipProvider, 607

    SSE user instancing, 577–582

    using file-based policy store, 768–771

connectionUsername attribute, 663

### containers

    AD and AD LDS, 640–641

    nesting, 660–662

    securing, 662–667

### content types, 30

context\_BeginRequest method, managed modules, 74

### ControlPrincipal permission, 192–193

<controls /> configuration section, AJAX, 798

### ControlThread permission, 192

ConvertByteArrayToString method, 403

ConvertStringKeyToByteArray method, 400, 403

### cookie domain, 332–333

### cookieless forms authentication, 308–323

    AutoDetect, 310–313

    for classic ASP, 391

    cookieless attribute, 309

    cookieless tickets/other URLs in pages, 317–319

    cross-application behavior, 335–338

    over non-SSL connections, 305

    overview of, 308–310

    payload size, 319–321

    replay attack, 315–316

    unexpected redirect behavior, 322–323

    UseDeviceProfile, 313–315

### cookieless sessions, 424–426

- CookiePath **property**, 355, 706
  - CookiePath **variable**, 345, 352
  - cookieProtection **attribute**, role cache cookies, 712
  - cookieRequiresSSL **attribute**, 711, 713
  - cookies**, 861–865
    - cookie domain, 332–333
    - cross-application behavior, 339–342
    - EndRequest, RoleManagerModule, 711–712
    - persistent, 288–291
    - role cache settings, 712–714
    - RoleManagerModule checks, 709
    - security best practices, 838–841
    - session-based, 288, 421–424
    - setting security options, 303–308
    - SSO-lite. See SSO-lite, cookie-based
    - tracking logon status, 419
    - WindowsTokenRoleProvider and, 728
  - cookieSlidingExpiration **attribute**, 711, 712
  - cookieTimeout **attribute**, 712
  - CreateDate **column**, aspnet\_Membership **table**, 575
  - createPersistentCookie **attribute**, 712
  - CreateUser **method**
    - ActiveDirectoryMembershipProvider, 651, 673
    - AuthorizationStoreRoleProvider, 766
    - overview of, 542–543
    - primary key and, 553
    - RoleProvider, 725
    - self-service password reset or retrieval, 548
    - SqlMembershipProvider, custom encryption, 596
    - SqlMembershipProvider, password history, 602, 605–606, 610–612, 617
    - SqlMembershipProvider, password strength, 598–600
    - tracking online users, 550
  - CreateUserwizard **control**
    - configuring self-service password resets, 672
    - SqlMembershipProvider, password history, 612
    - using application partition, 683–684
  - CreatingCookie event**, 805
  - CreationDate **property**, MembershipUser
    - AD and AD LDS directory schema mappings, 645
    - defined, 524
    - updatability of, 524
  - credentials**
    - AD and AD LDS connection settings, 645
    - designing Membership feature, 470
    - securing containers, 663–667
    - using application partition, 683
    - validating with MembershipProvider, 545–547
  - cross-application behavior**
    - cookieless, 335–338
    - redirects, 334–337
    - sharing of ticket, 333–334
  - cross-page postings, cookies**, 340
  - cross-site request forgery (CSRF or surf)**, 857–861
  - cross-site scripting threat**. See XSS (cross-site scripting threat)
  - cryptographic API (CAPI)**, 268
  - cryptographic service providers (CSPs)**, 268
  - cryptology, using custom Hash algorithms**, 558–559
  - <cryptographySettings /> **configuration element**, 261, 558–559
  - cspProviderName, 268
  - CSPs (cryptographic service providers)**, 268
  - CSRF (cross-site request forgery)**, 857–861
  - CSS**. See XSS (cross-site scripting threat)
  - CurrentConnectionProtection **property**, 654
  - CurrentPrincipal, Thread **class**
    - DefaultAuthentication, 117–120
    - handling asynchronous pipeline events, 109–110
    - locating security identity for requests, 87–92
    - summary review, 144–145
  - custom authentication service, AJAX 3.5**, 812–814
  - custom configuration classes**, 257–258
  - custom role service, AJAX**, 819
  - <customErrors> **configuration section, exception handling**, 861–862
  - CustomNativeHandler, 434
- ## D
- data protection API provider**. See DPAPIProtectedConfigurationProvider
  - database access**
    - encrypt data, 845–849
    - overview, 841
    - SQL Server, 843–845
    - Windows authentication, 842–843
  - database schema**, 562–573
    - ActiveDirectoryMembershipProvider mappings, 645–648
    - calling LOWER function, 572–573
    - common users table, 563–566
    - dbo user and, 586–588
    - installing with SQL Server Express. See SSE (SQL Server Express)
    - linking custom features to user records, 569–572
    - Membership, 573–577
    - overview of, 562
    - querying common tables with views, 568
    - SqlRoleProvider, 735–739
    - storing application name, 562–563
    - versioning provider schemas, 566–568
  - database security**
    - overview of, 584–586
    - for SQL, 445–446
    - SqlRoleProvider, 745
  - database-based policy store, SQL Server**, 780–783

## date-time

- DateTime assumptions, MembershipUser, 536–537
- issues with clock resets, 292–294
- setting for ticket expiration, 291–292
- dbo **users, and database schemas, 586–588**
- DCOM, 251–253**
- Decrypt **method**,
  - ProtectedConfigurationProvider, **275–276**
- decryption **attribute, 300**
- decryptionKey **attribute**
  - ASP.NET 2.0/3.5 encryption, 300
  - sharing tickets between ASP.NET versions, 324–325
  - SqlMembershipProvider, 589
- DecryptPassword **method**, MembershipProvider, **594–595**
- default.asp, **389–390**
- DefaultAuthenticationModule, **117–120**
- DefaultHttpHandler
  - ASP.NET 3.5, 383–384
  - not using in integrated mode, 388
  - passing data from ASP.NET to classic ASP, 411, 413
  - using with ASP.NET and classic ASP, 384–387
- defaultProvider **attribute**, Membership **class**, **520–521**
- DefaultProvider **property**, **498–499, 501–507, 515**
- Delegation of Control Wizard, 663–667**
- DeleteCookie **method**, Roles **class**, **693**
- DeleteRole **method**
  - AuthorizationStoreRoleProvider, 766
  - RolePrincipal/Roles classes, 694
  - RoleProvider, 725
- DeleteUser **method**
  - ActiveDirectoryMembershipProvider, 652
  - MembershipProvider, 543, 565
- deployment, IIS 7.0, 4–6**
- Description **column**, aspnet\_Roles **table**, **736–737**
- deserialization, session state and, 441–444**
- Design Patterns: Elements of Reusable Object-Oriented Software (Gamma, Helm, Johnson and Vlissides), 472**
- design-time API, 258–259, 277–278**
- device profiles, 313–315**
- DigestAuthenticationModule, **93**
- directory-based policy store, AzMan, 771–779, 785**
- DirectorySearcher **class**, **648–649**
- DirectoryServicesPermission, **685–688, 690**
- DNS namespace, 334**
- DnsPermission **class**, **188**
- DOMAIN/MACHINENAME\$, **584–585**
- DOMAIN/USERNAME, **security, 585**
- DOS (denial-of-service) attacks**
  - guarding against, 865–871
  - overview of, 825
  - session ID, 437–439
  - SqlMembershipProvider causing, 621

- DPAPIProtectedConfigurationProvider, **264–267**
  - defining in machine.config, 264
  - encrypting data, 845–849
  - keyEntropy option, 264–265
  - overview of, 259–260
  - selecting, 261–264
  - useMachineProtection option, 265–267
  - using in partial trust, 274–278
- dsacls.exe **tool**, **676**
- dsmgmt.exe **tool**, **676**
- DWORD **registry, OOP state server, 447**
- dynamic applications**
  - SqlMembershipProvider supporting, 626–632
  - SqlRoleProvider supporting, 757–758

## E

- e-commerce sites, cookieless session state and, 425–426**
- Edit Script Map dialog box, 377–378**
- elevation of privilege threat, 826**
- Email **column**, aspnet\_Membership **table**, **575**
- Email **property**, MembershipUser
  - AD and AD LDS directory schema mappings, 646, 648
  - defined, 524
  - updatability of, 524
- enableCrossAppRedirects, **cookie-based SSO-lite, 351**
- Enabled **property**, Roles **class**, **692**
- EnablePasswordReset **property**, **547, 650**
- EnablePasswordRetrieval **property**, **547, 654**
- enableSearchMethods **attribute**, **648, 654**
- EnableViewStateMac **attribute**, **pages, 451**
- encoding**
  - best practices, 832–834
  - protecting against cross-site scripting, 854–857
- Encrypt **method**, UserData **property**, **331**
- <EncryptedData /> **element**, **260**
- encryptedTicket **parameter**, RolePrincipal **class**, **704–705, 706–707**
- encryption, 592–594**
  - ASP.NET 2.0 and 3.5, 299–303
  - customizing SqlMembershipProvider, 594–597
  - database access best practices, 845–849
  - determining viewstate, 452–453
  - secure database access using, 845–849
  - session state cookie and, 423
  - SqlMembershipProvider, 588–590
- EncryptPassword **method**, MembershipProvider, **594–597**
- End **event handler**
  - asynchronous pipeline events, 101, 104–105, 107
  - asynchronous PreRender processing, 138–141
- EndInvoke **method**, **104**

EndProcessRequest, DefaultHttpHandler, **385–387**

EndRequest

forms authentication, 288

HTTP request processing, 143–144

RoleManagerModule, 711–712

**Enterprise level, CAS policies for, 162**

EnvironmentPermission **class, 188**

EvaluateIsValid( ) **method, 831**

**event validation, fraudulent postbacks, 460–462**

**evidence, code, 159**

**exception handling**

<customErrors> configuration section, 861–862

global, 864–865

IIS 7.0 troubleshooting improvements, 13

MembershipProvider class, 550–551

MembershipUser class, 528

migrating ASP.NET applications, 49

Try/Catch blocks, 863–864

**Execution permission, 192**

**-exp option, 272**

**expiration date, FormsAuthenticationTicket**

cookie-based SSO-lite, 353–355

enforcing, 291–294

enforcing single logins, 359, 362, 365–368

leveraging UserData property, 331

overview of, 289–290

sliding expirations, 308

Expired **property, RolePrincipal class, 706**

**expired sessions, 437–439**

ExpireDate **property, RolePrincipal class, 704, 706**

**extensions. See ISAPI extensions**

## F

**Façade pattern, provider model, 482–483**

**Factory Method pattern, provider model, 474–481**

**Failed Request Tracing feature, 12–16**

FailedPasswordAnswerAttemptCount **column, aspnet\_Membership table, 576, 618**

FailedPasswordAnswerAttemptWindowStart **column, aspnet\_Membership table, 576, 618**

FailedPasswordAttemptCount **column, aspnet\_Membership table, 576, 618**

FailedPasswordAttemptWindowStart **column, aspnet\_Member, 576, 618**

**Feature Delegation applet, IIS 7.0, 239–243**

**features**

linking to user records, 569–572

using aspnet\_Users table, 565

FileAuthorizationModule

authorization best practices, 828

overview of, 123–124

FileIOPermission

AuthorizationStoreRoleProvider in partial trust, 785

defining individual permissions, 154

design-time API and, 258–259

sandboxed access to ADODB using, 210–211

SqlRoleProvider in partial trust, 742–743

troubleshooting, 165–167

working with different trust levels, 153–154

FileIOPermission **class, 189**

**filtering data, before encoding, 832–834**

FindUsersByEmail **method**

ActiveDirectoryMembershipProvider, 648–649, 652

MembershipProvider, 545

FindUsersByName **method**

ActiveDirectoryMembershipProvider, 648–649, 652

MembershipProvider, 545

FindUsersInRole **method**

AuthorizationStoreRoleProvider not implementing, 766

RoleProvider, 726

SqlRoleProvider, 745

FindUsersInRoles **method, RoleProvider, 726**

**forms authentication, 287–372**

across different content types, 326–329

best practices, 827

for classic ASP with ASP.NET, 389–394

configuring in IIS 7.0, 323

enabling Role Manager with, 696

encryption in ASP.NET 2.0/3.5, 299–303

enforcing expiration, 291–294

enforcing logouts, 368–371

enforcing single logons, 358–368

leveraging UserData property, 329–332

overview of, 288

persistent tickets, 288–291

RoleManagerModule, 709

security of signed tickets, 295–299

session state security features vs., 419

setting cookie-specific security, 303–308

sharing tickets between 1.1 and 2.0/3.5, 324–325

**forms authentication, cookieless, 308–323**

AutoDetect, 310–313

cookieless tickets and other URLs in pages, 317–319

overview of, 308–310

payload size, 319–321

replay attack, 315–316

unexpected redirect behavior, 322–323

UseDeviceProfile, 313–315

**forms authentication, passing tickets across applications, 332–357**

cookie domain, 332–333

cookie-based SSO-lite. See SSO-lite, cookie-based

### forms authentication, passing tickets across applications (continued)

cross-application sharing of ticket, 333–342  
overview of, 332

FormsAuthenticationModule  
<authentication>, 39–40  
across different content types, 326–329  
authenticating requests, 115–117  
EndRequest, 144  
enforcing expiration, 291–294  
enforcing single logins, 363–365  
forms authentication tasks, 288  
native UrlAuthorizationModule with, 130

FormsAuthenticationTicket  
cookie-based SSO-lite. See SSO-lite, cookie-based  
enforcing expiration, 291–294  
enforcing single logins, 362–363  
leveraging UserData property, 329–332  
native UrlAuthorizationModule with, 130  
payload size with cookieless tickets, 319–321  
security of signed tickets, 295–299  
setting cookie timeout, 289–290  
sharing tickets between ASP.NET versions, 324–325  
unexpected redirect behavior, 322–323

FormsCookieName, **337**

FormsIdentity, **696**

**401 status code, 144**

**framework thread pool thread, 100–101**

**fraudulent postbacks, 458–462**

**Full Control ACLs, 250–251, 666**

**Full trust**  
configuring, 150  
defining, 148, 150  
intent of, 183  
working with, 151–153

**FullAccess role, database, 585–586, 745**

**FullTrust permission set, 157–158, 161**

## G

### GAC (Global Assembly Cache)

ActiveDirectoryMembershipProvider in partial trust, 688–689

AllowPartiallyTrustedCallersAttribute, 200–202

AuthorizationStoreRoleProvider in partial trust, 785

matching permission sets to code, 161

processRequestInApplicationTrust and, 217–221

sandboxing and, 204–211

session state and, 441–444

setting up configuration section for feature, 256

strongly named assemblies, APTCA and bin directory, 202–204

using Minimal trust, 149

gacutil tool, **200**

**GC (global catalog), AD, 640**

GeneratePassword method  
ActiveDirectoryMembershipProvider, 652  
Membership class, 523  
SqlMembershipProvider, 590–594

GenericPrincipal, **88–91, 115–117**

get\_defaultLoadCompletedCallback( ),  
**AJAX, 816**

get\_defaultLoginCompletedCallback( ),  
**AJAX, 807**

get\_defaultLogoutCompletedCallback( ),  
**AJAX, 807**

get\_isLoggedIn( ), **AJAX, 804, 807, 812**

get\_path( ) function, **AJAX, 806, 816**

get\_Roles( ) function, **AJAX, 816, 818**

GetAllRoles method, **726, 745**

GetAllUsers method,  
ActiveDirectoryMembershipProvider,  
**648–649, 652**

GetAllUsers method, MembershipProvider, **545**

GetNumberOfUsersOnline method, **549, 652**

GetPassword method,  
ActiveDirectoryMembershipProvider, **652**

GetPassword method, MembershipProvider,  
**548, 551**

GetRedirectUrl, cookie-based SSO-lite, **347–348**

GetRedirectUrl, leveraging UserData property,  
**331, 332**

GetRoles event, **708–709, 714–722**

GetRoles method, RoleManagerModule, **711**

GetRoles method, RolePrincipal class, **697–702, 704–705**

GetRolesForCurrentUser method, **AJAX, 815, 819**

GetRolesForUser method  
AuthorizationStoreRoleProvider, 766  
RolePrincipal/Roles classes, 694  
RoleProvider, 723, 724–725  
SqlRoleProvider, 745, 749–750, 752, 754  
WindowsTokenRoleProvider, 726–727, 729–730

GetSection method  
protected configuration providers, 276, 278  
reading and writing configuration, 244–245  
reading local configuration, 247–249  
run-time/design-time configuration APIs, 248

GetUser method, MembershipProvider  
overview of, 544  
primary key and, 553  
tracking online users, 550

GetUser method MembershipUser, **529**

getUserId stored procedures, **756**

GetUserInRole method, SqlRoleProvider, **751**

GetUserNameByEmail method, **544, 652**

GetUsersInRole method, **766**

GetUsersInRoles **method**, 726  
 GetWebApplicationSection **method**, 248  
**Global Assembly Cache**. *See* **GAC (Global Assembly Cache)**  
**global catalog (GC)**, AD, 640  
**global error handling**, 864–865  
 <globalModules /> **configuration section**, 34  
 Guid, 362–365  
**GUID identifier**, 755–757

## H

**Handler Mappings applet**, 376–377  
**Hash algorithms**, 556–560  
 hashAlgorithmType, 556–560, 588  
 HashAlgorithmType **property**, Membership, 521  
 HashPasswordForStoringInConfigFile **method**, 296  
**Health Monitoring feature**, 474  
 Helper **class**, hash, 405–410  
**hidden segments**, RequestFiltering **module**, 137  
**High trust**, 149–150, 153, 183  
**history**, implementing password, 602–617  
**HMACSHA1 algorithm**  
   passing sensitive data to classic ASP, 400–401, 405  
   signed tickets, 295–299  
**HTML images**, cross-site request forgery, 857–861  
 HtmlDecode **method**, HttpUtility **class**, 832  
 HtmlEncode **method**, HttpUtility **class**, 832, 854  
**HTTP request processing**  
   asynchronous page execution, 137–143  
   AuthenticateRequest, 110–117  
   AuthorizeRequest, 122–135  
   blocking requests at IIS level, 135–137  
   built-in IUSR account/IIS\_IUSRS group, 80–81  
   DefaultAuthentication and, 117–120  
   EndRequest, 143–144  
   integrated mode per-request security, 81–87  
   online resources, 98–100  
   OS thread identity, 92–98  
   overview of, 79  
   PostAuthenticateRequest, 120–122  
   PostAuthorizeRequest, 135  
   security identity, 87–92  
   session state in Integrated mode, 427–428  
   thread identity/asynchronous pipeline events, 100–110  
   Thread.CurrentPrincipal and, 117–120  
 HttpCachePolicy, 384  
 HttpContext.Current.User **property**, 111–117  
 <httpErrors /> **configuration section**, 237  
 HttpHandler. *See also* **managed handlers**  
   configuring identity, 48–49  
   extending IIS 7.0 with, 236  
   overview of, 45–48

  passing data to ASP, 392–394  
   registering HTTP handlers in, 386–387  
   review, 78  
 <httpHandlers /> **configuration section**, **AJAX**, 798–799  
 HttpModule. *See also* **managed modules**  
   authenticating requests and, 110–117  
   configuring identity, 48–49  
   defined, 30  
   developing managed modules, 67–69  
   extending IIS 7.0 with, 236  
   Init method, 73–74  
   overview of, 43–45  
   review, 78  
   session state in Integrated mode, 427–434  
 <httpModules /> **configuration section**, **AJAX**, 799  
 HttpOnly **cookies**  
   overview of, 306–308  
   protecting session cookies, 423–424  
   role cache cookies, 713–714  
   security best practices, 839  
 HttpRuntime **object**, 742  
 HttpUtility **class**, 185  
**Hyper Text Protocol Stack**, 19

## I

IAAsyncResult **interface**, 104–105  
 IAzApplicationContext **interface**, 764–765  
**identity**. *See also* **thread identity**  
   asynchronous page execution, 137–143  
   authentication best practices, 827–828  
   configuring AnonymousAuthenticationModule, 85–86  
 <identity /> **configuration section**  
   database security, 584–586  
   establishing OS thread identity, 93–97  
   integrated mode per-request processing, 91  
   overview of, 48–49  
   thread identity/asynchronous pipeline events, 106  
   using stores depending on user identity, 263–264  
   WindowsAuthenticationModule and, 111  
 Identity **property**, RolePrincipal **class**, 698  
 IHttpHandler **interface**. *See* **managed handlers**  
 IHttpModule **interface**. *See* **managed modules**  
 IIdentity **reference**, 696–697, 709  
**IIS (Internet Information Services) 5.0**, ISAPI in, 374–375  
**IIS (Internet Information Services) 7.0**  
   application pools, 17–19  
   ASP.NET integration, 9–10  
   components, 19–21  
   configuring session state inside, 426–427  
   deployment and configuration management, 4–6  
   improved administration, 6–9

## **IIS (Internet Information Services) 7.0 (continued)**

- managing application's roles through, 758–760
- managing ASP.NET configuration vs., 233–235
- modular architecture, 2–3
- modules, 22–26
- security improvements, 11–12
- summary review, 26–27
- troubleshooting improvements, 12–17
- wildcard mappings. See wildcard mappings, IIS 7

## **IIS (Internet Information Services) 7.0, Integrated mode, 29–78**

- advantages of, 30–31
- architecture overview, 31–34
- authenticating classic ASP, 394–395
- authorization configuration section, 40–42
- authorizing classic ASP, 410–414
- configuring session state, 427–435
- improvements to, 9–10
- managed handlers. See managed handlers
- managed modules. See managed modules
- migrating ASP.NET applications to, 42–49
- security configuration section group, 38–40
- serving classic ASP in, 387–388
- system.webServer configuration section group, 34–37

## **IIS Manager**

- configuring impersonation, 95–98
- default identity of application pools, 92
- editing <forms /> authentication, 323
- Failed Request Tracing, 12–13
- installing managed handler, 67
- installing managed module, 77
- overview of, 6–8
- security improvements, 12

IIS\_IUSRS **group**, **12, 80–81**

IIS\_USR, **12**

IIS\_WPG **group**, **80**

IisTraceListner **class**, **16–17**

ildasm **utility**, **199–200**

<IMembershipCondition /> **element**, **160**

## **impersonation**

- configuring ASP.NET, 95–98
- establishing OS thread identity, 92–95
- WindowsAuthenticationModule, 114–115

**information disclosure threat**, **825**

**information leakage threat, AJAX-enabled**, **871–874**

Init **method**, IHttpModule, **68–69, 73–74**

Initialize **method**

- ActiveDirectoryMembershipProvider, 688
- building provider-based feature, 499–504, 513–514, 517
- ProviderBase, 485
- Singleton pattern, 481–482
- SqlMembershipProvider, 606–607, 622–623
- SqlRoleProvider, 739

**in-process session state**, **420–421**

**input data, validating**. See **user input validation, best practices**

InsertHistoryRow **method**,  
SqlMembershipProvider, **605–608, 610–611, 614–617**

InstallCommon.sql **file**, **562**

InstallMembership.sql, **573–576**

InstantiateProviders, **Factory Method pattern**,  
**475–480**

**Integrated mode application pool**, **18, 26**

**Integrated mode, ASP.NET**, **10–11**

**Internet Explorer**, **312–313**

**Internet Information Services**. See **IIS (Internet Information Services) 7.0**

<IPermission /> **element**

- customizing OleDbPermission, 174
- customizing OleDbPermission, 172–173
- defining individual permissions, 159
- enabling new permissions in policy file, 168–171
- using WebPermission, 178

IPrincipal **interface**, **695–696, 709**

**IPSEC (IP Security), OOP server**, **447**

IRequestSessionState **interface**, **427–428**

IsAccessibleToUser, SiteMapProvider,  
**463–464, 467**

IsAnonymous **column**, aspnet\_Users **table**, **566**

## **ISAPI extensions**

- behavior in IIS 5, 374–375
- configuring wildcard mapping, 377–380
- new extensibility API vs., 3
- overview of, 31–32

IsApproved **column**, aspnet\_Membership  
**table**, **575**

IsApproved **property**,

ActiveDirectoryMembershipUser, **645, 655**

IsApproved **property**, MembershipUser, **524**

IsCurrentUserInRole **method**, RoleService, **815**

IsCurrentVersion **column**, provider schemas,  
**567–568**

ISerializable **method**, **442–444**

IsInRole **method**, RolePrincipal **class**, **695, 697–700, 705**

IsLockedOut **column**, aspnet\_Membership  
**table**, **575**

IsLockedOut **property**,

ActiveDirectoryMembershipUser, **645, 655–656**

IsLockedOut **property**, MembershipUser, **525, 535**

IsLoggedIn **method**, AuthenticationService, **805**

IsolatedStorageFilePermission **class**, **189–190**

IsOnline **property**, MembershipUser, **524–525, 528**

IsReusable **method**, IHttpHandler, **51, 60**

IsReusable **property**, session state, **432**

IsRoleListCached **property**, RolePrincipal, **698**

IssueDate **property**, 331, 706  
 IsUserInRole( ) **function**, AJAX, 816, 818  
 IsUserInRole **method**  
   AuthorizationStoreRoleProvider, 766–767  
   RolePrincipal/Roles classes, 694  
   RoleProvider, 723, 724–725  
   SqlRoleProvider, 745, 749–754  
   WindowsTokenRoleProvider, 726–732  
 IsValid **property**, Page, 831  
 IUSR **built-in account**, 85–87  
   FileAuthorizationModule, 123–124  
   overview of, 80–81

## J

**JavaScript and Object Notation (JSON)**, 796, 874–875  
**JSON (JavaScript and Object Notation)**, 796, 874–875

## K

keyContainerName  
   RSA provider, 268, 269–271  
   synchronizing key containers, 272–273  
 keyEntropy,  
   dpapiProtectedConfigurationProvider,  
   264–265  
 keys  
   encryption in ASP.NET 2.0 and 3.5, 299–301  
   generating programmatically, 301–303

## L

**Language Integrated Query (LINQ)**, 179–181  
 LastActivityDate **column**, aspnet\_Users **table**,  
   565–566  
 LastActivityDate **property**, MembershipUser,  
   525, 549–550, 646, 654  
 LastLockOut **property**, MembershipUser, 646  
 LastLockoutDate **column**, aspnet\_Membership  
   **table**, 575  
 LastLockoutDate **property**, MembershipUser,  
   525, 620  
 LastLoginDate **column**, aspnet\_Membership  
   **table**, 575  
 LastLoginDate **property**, MembershipUser, 525,  
   646, 654  
 LastPasswordChangedDate **column**,  
   aspnet\_Membership **table**, 575  
 LastPasswordChangedDate **property**,  
   MembershipUser, 525  
**LDAP (Lightweight Directory Access Protocol)**  
   ActiveDirectoryMembershipProvider, 640,  
   642–645  
   AzMan support for query groups, 778–779

**LDIF files**, 678–679  
 Level **property**, AspNetHostingPermission, 182  
 LinkDemand **exception behavior**, 185, 195–198,  
   202–203  
**LINQ (Language Integrated Query)**, 179–181  
**listener adapters**, 19–21  
 load( ) **function**, AJAX, 816–817  
 Load **event**, cookie-based SSO-lite, 354–355  
**Load Roles button**, 817  
 loadCompletedCallback( ) **function**, AJAX, 817–818  
 LocalSqlServer, 583–584  
 <location /> **configuration element**  
   allowOverride attribute, 226  
   IIS 7.0 feature delegation, 240–243  
   locking provider definitions, 233  
   overview of, 223–225  
   path attribute, 225–226  
   UrlAuthorizationModule, 127–128  
**lock attributes**, 227–233  
   finding available elements, 229  
   locking attributes, 227–229  
   locking elements, 229–231  
   locking provider definitions, 231–233  
   using, 227  
 lockAllAttributesExcept, 227–229  
 lockAllElementsExcept, 227, 229–231  
 lockAttributes, 227–229, 243  
**locked-down mode**, IIS 7.0, 11  
 lockElements, 227, 229–231  
**locking attributes**, 243  
**lockouts**, account  
   implementing automatic unlocking, 621–626  
   overview of, 620  
   SqlMembershipProvider, 617–621  
 LoggedIn **event**, single logins, 360, 362, 367  
 LoggingIn **event**, single logins, 360, 362, 367–368  
 LoggingOut **event**, enforcing logouts, 369–371  
**login**. *See also* SSO-lite, cookie-based  
   AJAX, 808–812  
   classic ASP authentication, 389–391  
   database security, 585  
   enforcing single, 358–368  
   MembershipProvider, 551  
   replay attacks and, 315–316  
   session state and, 417–420  
   Windows authentication, 112–113, 842–843  
 login( ) **function**, AuthenticationService, 806  
 Login **method**, AJAX, 805, 812–814  
**logout**  
   AJAX, 812  
   enforcing, 358, 368–371  
 logout( ) **function**, AuthenticationService, 806  
 Logout **method**, AJAX, 805, 812–814  
 LogRequest, 100

## Low trust

- defining, 149–150
  - enabling `SqlRoleProvider` in, 739
  - intent of, 183
  - Membership requirement, 553
  - working with, 154
- `LoweredEmail` **column**, `aspnet_Membership` **table**, 575
- `LoweredRoleName` **column**, `aspnet_Roles` **table**, 736–737
- `LOWER()` **function**, 572–573

## M

### Machine level, CAS policies, 162

- `machine.config` **file**
- IIS 7.0 configuration, 235
  - Membership class, 520
  - protected configuration providers, 264
  - `requirePermission` attribute, 256–257
- `<machineKey />` **element**
- `compatibilityMode` attribute, 454
  - encryption in ASP.NET 2.0 and 3.5, 299–301
  - passing sensitive data to classic ASP, 399–400
  - sharing tickets, 324–325
  - signed tickets, 295–296
  - `SqlMembershipProvider` encryption, 588–590
  - viewstate protection, 451–454

### Managed Engine, 387–388

#### managed handlers, 60–67

- defined, 50
- developing, 51–60
- `DisplayEmployee` method, 63–67
- extending IIS 7.0 with, 236
- installing, 67
- `IsReusable` method, 60
- native configuration systems vs., 236–238
- overview of, 50–51
- `ProcessRequest` method, 60–63
- summary review, 77–78

#### managed modules, 67–77

- `context_beginRequest` method, 74–77
  - developing, 69–73
  - extending IIS 7.0 with, 236
  - forms authentication using, 326–329
  - `Init` method, 73–74
  - installing, 77
  - native configuration systems vs., 236–238
  - overview of, 25–26, 67–69
  - passing data from ASP.NET to classic ASP with, 411–414
  - summary review, 77–78
  - `UrlAuthorizationModule`, 124–129
- `ManagedHandler`

- applications running in session state, 432–435
  - authenticating classic ASP, 395
  - session state in IIS 7.0 Integrated mode, 432, 434
- `MapRequestHandler`, 100, 428

- `MaxCachedResults` **property**, 693, 712
- `maxInvalidPasswordAttempts` **attribute**, 546, 618–620

- `.mdf` **file**, 582–583

#### Medium trust

- defining, 149–150
- finding policy file, 155–156
- intent of, 183
- LINQ in applications of, 179–181
- restrictions of, 184

#### membership, ASP.NET, 792–794

- Membership **class**, 520–523, 792
- membership condition, defined, 159
- Membership feature, 519–560. **See also** AJAX 3.5

- `ActiveDirectoryMembershipProvider`, 649–650

- authenticating classic ASP with ASP.NET, 389

- custom Hash algorithms with, 556–560

- database schema, 573–577

- database security, 586

- designing for credentials, 470–472

- enforcing single logins, 359–368

- Facade pattern, 482–483

- Factory Method pattern, 475–480

- Membership class, 520–523

- `MembershipProvider`.

- `See` `MembershipProvider` class

- `MembershipUser`. `See` `MembershipUser` class

- overview of, 358–359, 519

- primary key for, 552–553

- Role Manager with, 786–789

- Strategy pattern, 473

- summary review, 560

- supported environments, 553–556

- updating `LastActivityDate`, 565

- `MembershipPasswordException` **type**, 551

- `MembershipProvider` **class**, 537–551

- ASP.NET membership, 792

- custom password encryption, 594–597

- error-handling, 550–551

- overview of, 537–541

- retrieving data for single/multiple users, 544–545

- self-service password reset/retrieval, 547–549

- tracking online users, 549–550

- user creation/updates, 541–544

- validating user credentials, 545–547

- `MembershipUser` **class**, 523–537

- `ActiveDirectoryMembershipUser`, 654–657

- AD and AD LDS directory schema mappings, 645–648

- ASP.NET membership, 792

- `DateTime` assumptions, 536–537

enforcing single logins, 359–363, 367–369  
 extending, 525–529  
 overview of, 523–525  
 updates and, 529–535

#### methods

ActiveDirectoryMembershipProvider, 651–654  
 AuthorizationStoreRoleProvider, 766–767  
 Membership class, 521–523  
 MembershipProvider, 542–543  
 RolePrincipal class, 697–698  
 RoleProvider, 724–726  
 Roles class, 693–694  
 self-service password reset or retrieval, 548–549  
 System.Web.ApplicationServices  
   .AuthenticationService class, 805  
 System.Web.ApplicationServices  
   .AuthenticationServices.RoleService  
   class, 815  
 validating user passwords, 546–547

#### Microsoft Mobile Internet Toolkit (MMIT), 303

MicrosoftAjax.js, 801–802

MicrosoftAjaxWebForms.js, 801

Microsoft.Web.Administration API, 7–8

migrating, ASP.NET to Integrated mode, 42–49

#### Minimal trust

defined, 149–150  
 intent of, 184  
 working with, 154–155

minRequiredNonAlphanumericCharacters  
 property

ActiveDirectoryMembershipProvider, 650  
 MembershipProvider, 541–542  
 SqlMembershipProvider, 590–591, 598–617  
 minRequiredPasswordLength property  
 ActiveDirectoryMembershipProvider, 650  
 MembershipProvider, 541  
 SqlMembershipProvider, 591, 598–617

Microsoft Anti-Cross Site Scripting Library, 855–857

MMIT (Microsoft Mobile Internet Toolkit), 303

mobile users, cookieless session state for, 425–426

MobileAlias column, aspnet\_Users table, 566

MobilePIN column, aspnet\_Membership table, 574

modular architecture, IIS 7.0, 2–3

#### modules

architecture, 2–3  
 ASP.NET Integrated mode and, 30  
 developing, 3  
 managed, 25–26  
 overview of, 22  
 unmanaged, 22–25

<modules> configuration section

ASP.NET Integrated mode architecture, 35–37  
 installing managed modules with, 77  
 passing data from ASP.NET to classic ASP, 411

## N

name attribute, <modules /> configuration section, 35  
 native modules

managed configuration systems vs., 236–238  
 UrlAuthorizationModule, 129–135  
 using forms authentication, 326–329

nesting, supported by AzMan, 776–778

nesting containers, 660–662

.NET Framework Configuration MMC, 456–457

.NET Roles applet, 758–760

.NET Users applet, 632–636

#### NETWORK SERVICE account

authentication best practices, 827–828  
 database security and, 584–586  
 installing ADLDS with application partitions, 679  
 securing containers, 662–667  
 SQL Server Express, 578, 580–583  
 Windows authentication, 843

no-compile pages, 215–217

Nothing permission set, 158, 160

NotSupportedException, MembershipProvider,  
 550

NTFS ACLs, 578–579

## O

OAEP (Optional Asymmetric Encryption and  
 Padding), 268

OdbcPermission, customizing, 173–176

oidgen.exe, 668

OleDbPermission

customizing, 171–173

enabling new permissions in policy file, 169–170  
 reducing security by allowing, 175–176

OnExecuteUrlPreconditionFailure,  
 DefaultHttpHandler, 385

OnFailed() function, AJAX, 811

#### online resources

Anti-Cross Site Scripting Library, 857  
 ApplicationHost.config file, 6  
 DPAPI provider, 845  
 extending Membership provider model, 523–524  
 IIS 7.0 modules and features, 3  
 Microsoft.Web.Administration API, 8  
 migrating ASP.NET applications, 49  
 RequestFiltering, 12, 137  
 Required Access Control Lists, 828  
 RSA provider, 845  
 SQL Injection Cheat Sheet, 850  
 tracing system, 17  
 validating query strings, 838  
 WCF listener adapters/hosting WCF in IIS 7.0, 21

online users, tracking, 549–550

OnLogin() function, AJAX, 810

# OnLoginCompleted() function

---

OnLoginCompleted() **function, AJAX, 811**

OnLogout() **function, AJAX, 812**

OnRolesLoaded() **function, AJAX, 818**

**OOP (out-of-process) state server**

  securing, 447

  session data partitioning and, 420–421

  session ID DoS attacks and, 437–438

Open\* **methods, 248, 249**

OpenWebConfiguration **method, 249**

**operating system, establishing thread identity, 92–98, 117**

**Optional Asymmetric Encryption and Padding (OAEP), 268**

\$OriginHost\$ **string replacement token, 157, 176**

out **parameter, MembershipProvider, 550**

OutputCacheModule, **237**

OverrideExecuteUrlPath, DefaultHttpHandler, **385–387**

**overrides, MembershipUser, 528**

## P

-pa **switch, 270–271, 273**

Page\_Error **event, global error handling, 865**

PageAsyncTask, **141–143**

pageLoad() **function, AJAX, 810**

**pages and compilation, 448–468**

  fraudulent postback problem, 458–462

  page compilation, 454–457

  request validation, 450–451

  site navigation security, 462–467

  summary review, 468

  viewstate protection, 451–454

Page.Validate **method, user input, 831**

**parameters, preventing SQL injection, 851–852**

**partial trust**

  APTCA requirements, 198–204

  AuthorizationStoreRoleProvider in, 783–785  
  configuration, 253–259

  defining, 148–149

  exception behavior in Link demands, 195–198

  LINQ applications in, 179–181

  protected configuration, 275–277

  sandboxing access to security sensitive code,  
  204–211

  SqlRoleProvider in non-ASP.NET, 739–745

  using ActiveDirectoryMembershipProvider in,  
  684–689

  working with

    processRequestInApplicationTrust,  
    214–221

**partitioning, session data, 420–421**

PassingDataToClassicASP, **411–414**

Password **column, aspnet\_Membership table, 574, 576**

**password salts, 602**

PasswordAnswer **property, 575, 648**

passwordAnswerAttemptLockoutDuration  
  **attribute, 650, 654**

passwordAttemptWindow **attribute**

  ActiveDirectoryMembershipProvider, 650

  MembershipProvider, 546

  SqlMembershipProvider, 618–619

passwordFormat **attribute**

  ActiveDirectoryMembershipProvider, 654

  MembershipProvider, 548

  SqlMembershipProvider, 589

PasswordFormat **column, aspnet\_Membership table, 574**

PasswordHistory **table, 602–604, 612–613, 616–617**

PasswordQuestion **column, aspnet\_Membership table, 575**

PasswordQuestion **property, MembershipUser**

  AD and AD LDS directory schema mappings, 646, 648  
  defined, 525

  MembershipUser state after updates, 529–530

  updatability of, 535

PasswordRecovery **control, 626, 672**

**passwords,**

  ActiveDirectoryMembershipProvider,  
  **645–648, 667–675**

**passwords, MembershipProvider**

  creating/updates for, 541–544

  supporting self-service reset or retrieval, 547–549

  validating user, 545–547

**passwords, SqlMembershipProvider**

  changing formats, 588–590

  customizing generation of, 590–594

  enforcing custom strength rules, 598–617

PasswordSalt **column, aspnet\_Membership table, 574, 576**

passwordStrengthRegularExpression **attribute**

  ActiveDirectoryMembershipProvider, 650

  MembershipProvider, 542

  SqlMembershipProvider, 598–617

PasswordUsedBefore **method,**

  SqlMembershipProvider, **612–615**

path **attribute, <location /> element**

  IIS 7.0 feature delegation, 241

  overview of, 225–226

  payload size with cookieless tickets, 320–321

**path credentials, security configuration, 82–83**

**patterns, provider model, 472–483**

  Façade pattern, 482–483

  Factory Method pattern, 474–481

  Singleton pattern, 481–482

  Strategy pattern, 472–474

**payload size, cookieless tickets, 319–321**

PerformCentralLogin **method, cookie-based SSO-lite, 348–351**

**permission sets**

- creating custom trust levels, 168–171
- customizing `OdbcPermission`, 173–175
- customizing `OleDbPermission`, 171–173
- defining individual permissions, 158–159
- matching to code, 159–161
- overview of, 157–158
- sandboxed access to ADODB, 209–211
- sandboxed assemblies asserting, 206–208
- `SqlRoleProvider` in partially trusted non-ASP.NET, 742–744
- troubleshooting complex, 165–167
- using `WebPermission`, 176–179

**permissions**

- configuring partial trust, 253–259
- database security, 585
- reading local configuration, 247–249
- remote editing, 251–253
- running applications with minimum, 829
- securing containers, 663–667
- trust levels and session state, 439–441
- writing local configuration, 249–251

**permissions, default security, 181–195**

- `AspNetHostingPermission`, 182–187
- `DnsPermission` class, 188
- `EnvironmentPermission` class, 188
- `FileIOPermission` class, 189
- `IsolatedStorageFilePermission` class, 189–190
- `PrintingPermission` class, 190
- `ReflectionPermission` class, 190–191
- `SecurityPermission` class, 192–193
- `SmtpPermission` class, 193
- `SocketPermission` class, 193–194
- `WebPermission` class, 194
- `ZqlClientPermission` class, 194

**PermitOnly method**

- demanding permissions from configuration class, 257–258
- `processRequestInApplicationTrust`, 217, 221
- session state, 441, 444

**per-request security, Integrated mode, 81–87****persistent cookies, 712****persistent tickets, 288–291****personalization**

- Facade pattern, 483
- Strategy pattern, 473–474

**phishing attacks, 358****Physical Path Credentials field, IIS Manager, 82–83****-pi command, 273****poisoning, 437****policy files**

- creating for custom trust level, 167
- finding, 155–156
- permission sets, 157–161

string replacements, 156–157

working with, 162–165

**policy store, AzMan**

- deploying, 764–765
- directory-based, 771–779
- file-based, 768–771
- SQL Server database-based, 780–783

**ports, OOP state server, 447****POST requests, ASP.NET AJAX 3.5, 796**

`PostAcquireRequestState` event, 429, 432–434

`PostAuthenticateRequest` event

- asynchronous pipeline events, 101–109
- HTTP request processing, 120–122
- passing data from ASP.NET to classic ASP, 414
- `RoleManagerModule`, 120–122, 707–711
- setting cookie timeout, 290

`PostAuthorizeRequest` event

- HTTP request processing, 135
- passing data from ASP.NET to classic ASP, 414
- using `PreRequestHandlerExecute`, 135

**postbacks, fraudulent, 458–462**

`PostDeserialize` method, 257–258

`PostLogRequest`, 100

`PostMapRequestHandler`, 428–429, 432–433

**precompilation, 454–457**

precondition attribute, 35, 395

`PreRender` processing, asynchronous, 138–141

`PreRequestHandlerExecute`, 135

**-pri option, 272**

`PrintingPermission` class, 190

**privileges**

- best practices, 829
- elevation of privilege threat, 826

**process manager, Windows Process Activation Service, 20–21**

`<processModel />` element, 261, 263–264

`ProcessRequest` method, `IHttpHandler`, 51, 60–63

`processRequestInApplicationTrust`, 214–221, 441

**Professional IIS 7 and ASP.NET Integrated Programming (Wrox), 8, 67****Profile**

- Facade pattern, 483
- Strategy pattern, 473
- updating `LastActivityDate`, 565–566

**properties**

- `AuthorizationStoreRoleProvider`, 767
- `Membership` class, 521
- `MembershipProvider`, 541–542
- `MembershipUser` class, 524
- `RolePrincipal` class, 698, 706
- `Roles` class, 692–693
- self-service password reset or retrieval, 547–548
- validating user passwords, 546

## protected configuration

- defined, 259
- DPAPI provider, 264–267
- overview of, 259–260
- providers, 264
- redirecting with custom provider, 278–285
- RSA provider. See `rsaProtectedConfigurationProvider`
- selecting provider, 260–264
- using in partial trust, 275–277
- what you cannot protect, 260

## protocol listeners, 19

## provider definitions, locking, 231–233

## provider model, 469–517

- building provider-based feature, 495–517
- extending Membership, 523–524
- Facade pattern, 482–483
- Factory Method pattern, 474–481
- reasons for, 469–472
- Singleton pattern, 481–482
- Strategy pattern, 472–474
- summary review, 517
- `System.Configuration` classes, 490–494
- `System.Configuration.Provider` classes, 484–489
- `System.Web.Configuration` classes, 489

## `ProviderBase` class, 484–485

## `ProviderCollection` class, 486–488

## `ProviderException` class, 485–486, 551

## `ProviderName` property, 524, 528, 698

## providers, Membership class, 520–521

## providers, Role Manager

- `AuthorizationStoreRoleProvider`. See `AuthorizationStoreRoleProvider`
- `AuthorizationStoreRoleProvider`
- `SqlRoleProvider`. See `SqlRoleProvider`
- `WindowsTokenRoleProvider`. See `WindowsTokenRoleProvider`
- `ProviderSettings` class, 490–492
- `ProviderSettingsCollection` class, 492–494
- `ProvidersHelper` class, 475–476, 489, 504
- `Provider(s)` property, 506–507, 521, 692
- `ProviderUserKey` property, `MembershipUser`
  - `ActiveDirectoryMembershipUser`, 655–656
  - AD and AD LDS directory schema mappings, 645
  - creating/retrieving users, 553
  - defined, 524
  - linking custom features to user records, 569
  - updatability of, 524, 534

## Publish Website, 454–455

# R

## reading configuration, 244–253

- overview of, 244–247
- permissions for local, 247–249
- permissions for remote editing, 251–253

## read-only validation, `ProviderCollection`, 488

## read-only web servers, 149

## Read/Write delegation, features, 240

## `RedirectFromLoginPage`, forms authentication

- `AutoDetect?cookieless` option, 312
- cross-application redirects, 334–335
- `HttpOnly` cookies, 306
- leveraging `UserData` property, 331
- persistent tickets, 289–290
- setting cookies, 304–305

## redirects

- cookieless forms authentication, 322–323
- cross-application, 334–335
- `Reflection` namespace, `LinkDemand`, 198
- `ReflectionPermission` class, 181, 190–191
- `regenerateExpiredSessionId` attribute, 436
- `RegisterAsyncTask` method, 142
- `RegisterRequiresViewStateEncryption` method, `Page` class, 453
- `RegisterScripts` method, `ScriptManager`, 803
- `RegisterUniqueScripts` method, `ScriptManager`, 803

## `RegistryPermission` class, 191

## remote connections, using IIS Manager, 12

## remote editing, permissions for, 251–253

## RemotingConfiguration permission, 192

## `RemoveUserFromRole` method, 694

## `RemoveUserFromRoles` method, 694

## `RemoveUsersFromRole` method, 694

## `RemoveUsersFromRoles` method

- `AuthorizationStoreRoleProvider`, 767
- `RolePrincipal/Roles` classes, 694
- `RoleProvider`, 725
- `SqlRoleProvider`, 738

## replay attacks

- with cookieless tickets, 315–316
- preventing during logout, 369

## ReportingAccess role, database, 585–586, 745

## repudiation threat, 826

## request validation, 450–451, 835

## `RequestFiltering` module, 11–12, 135–137

## `Request.Form` (“key”), cross-site request forgery, 858–860

## requests per second (RPS), and DOS attacks, 439

## Required Access Control Lists, 828

## `requirePermission` attribute, 255–257

## `RequiresQuestionAndAnswer` attribute, 547, 650

## `requireSSL` attribute, forms authentication

- cookieless cross-application behavior, 338
- cookieless tickets and, 317
- tickets issued in cookies, 303–305

## `requireUniqueEmail` attribute, 542, 649

## `ResetPassword` method

- `ActiveDirectoryMembershipProvider`, 653, 673
- `MembershipProvider`, 549, 551

- SqlMembershipProvider, 598–600, 605–606, 617, 625–626
  - resourceType, **wildcard mappings**, 382–383
  - Response.Redirect **call**, 331
  - RestrictedMemberAccess **permission**, LINQ, 179–181
  - ReturnUrl **variable**, **cookie-based SSO-lite**, 352, 355–356
  - reuse, **session ID**, 435–436
  - RevertAssert **method**, CodeAccessPermission, 210–213
  - Rijndael algorithm**, 840–841
  - RijndaelManaged **class**, **AES**, 300
  - role cache**, RoleManagerModule, 712–714
  - Role Manager**, 691–733
    - AuthorizationStoreRoleProvider. See AuthorizationStoreRoleProvider
    - authorizing classic ASP with ASP.NET, 396
    - Facade pattern, 483
    - overview of, 691
    - passing user roles to classic ASP, 397–398
    - RoleManagerModule. See RoleManagerModule
    - RolePrincipal class, 695–707
    - RoleProvider class, 722–726
    - Roles class, 692–695
    - SqlRoleProvider. See SqlRoleProvider
    - Strategy pattern, 473
    - summary review, 732
    - using Membership together with, 786–789
    - WindowsTokenRoleProvider, 726–732
  - RoleExists **method**
    - AuthorizationStoreRoleProvider, 767
    - RoleProvider, 725
    - SqlRoleProvider, 745
  - <roleManager> **configuration section group**, 794
  - RoleManagerModule, 707–722
    - EndRequest, 711–712
    - multiple providers during GetRoles, 714–722
    - overview of, 707
    - PostAuthenticateRequest, 120–122, 707–711
    - role cache cookie settings and behavior, 712–714
    - SqlRoleProvider in Windows authentication, 746–747
    - summary review, 732–733
  - RolePrincipal **class**
    - ASP.NET role management, 794–795
    - multiple providers during GetRoles, 717–720
    - overview of, 695–707
    - RoleManagerModule, 709, 711–712
    - Roles class interacting with, 694
    - SqlRoleProvider in Windows authentication, 747
    - summary review, 732–733
  - RoleProvider **class**
    - ASP.NET role management, 794–795
    - overview of, 722–726
  - SqlRoleProvider. See SqlRoleProvider
  - WindowsTokenRoleProvider. See WindowsTokenRoleProvider
  - roles. **See also Role Manager**
    - configuring native UrlAuthorizationModule, 131
    - database security using, 585–586
    - managing AJAX. See AJAX 3.5
    - managing in ASP.NET, 794–796
    - nesting, 724
    - site navigation security, 463–465
  - Roles **class**, 692–695, 794
  - RoleService **class**, 815–816
  - RoleServiceManager **class**, 803–804
  - RPS (requests per second)**, and **DOS attacks**, 439
  - RsaProtectedConfigurationProvider, 267–273
    - aspnet\_regiis options, 273–274
    - defining in machine.config, 264
    - encrypting data, 845–849
    - keyContainerName, 269–271
    - overview of, 259–260
    - in partial trust, 274–278
    - selecting, 261–264
    - synchronizing key containers across machines, 272–273
    - useMachineContainer, 271–272
  - RSCA (Runtime Status and Control) API**, 17
  - runtime**, **no protected configuration for**, 261
  - Runtime Status and Control (RSCA) API**, 17
- ## S
- SAMAccountName **attribute**
    - ActiveDirectoryMembershipProvider, 659–660, 689
    - using Role Manager with membership, 786, 788
  - sandboxing**, 3–4
  - sandboxing**, **with strongly named assemblies**
    - access to ADODB, 208–211
    - access to SqlClientPermission, 212–214
    - access to System.Data.SqlClient, 212–214
    - overview of, 204–208
  - sanitization**, 832–833
  - SanitizeData **method**, 833
  - schema**. **See database schema**
  - ScopeName **property**,
    - AuthorizationStoreRoleProvider, 767
  - ScriptManager **control**, 801, 803–804
  - SDDL (Security Descriptor Definition Language)**, 656
  - search settings**
    - ActiveDirectoryMembershipProvider, 648–649
    - retrieving data for multiple users, 544–545
    - retrieving data for single user, 544
  - Secure **property**, **secure cookies**, 839

## **Secure Socket Layer. See SSL (Secure Socket Layer)**

### **security. See also identity; thread identity**

- container, 662–667
- database, 584–588
- default permissions. See permissions, default security
- HTTP requests, 87–92
- IIS 7.0 improvements to, 11–12
- Integrated mode, per-request, 81–87
- long-lived form authentication tickets, 289
- partial trust and no-compile pages, 215–217
- reducing using ODBC and OLEDB, 175–176
- RequestFiltering and, 137
- signed tickets, 295–299
- SqlRoleProvider, 739–745
- SSO-lite solution, 358
- trust levels. See trust levels
- web application. See ASP.NET security, web application best practices

### `<security />` **configuration section group**

- `<authentication>` configuration section, 38–40
- `<authorization>` configuration section, 40–42

## **Security Descriptor Definition Language (SDDL), 656**

### **security identifiers (SIDs),**

- WindowsTokenRoleProvider, **727–728**
- security trimming, SiteMapProvider, 462–464**
- `<SecurityClass />` **element**
  - registering DirectoryServicesPermission, 206
  - registering OleDbPermission, 169–170, 173
  - registering permission classes, 206

### SecurityIdentifier **class, 654–657**

### SecurityPermission **class, 192–193, 210–211**

### securityTrimmingEnabled **attribute, 462–464**

### SelectingProvider **event, 815**

### self-service password resets, **667–675**

### sensitive data, safely passing to classic ASP, **398–406**

### serialization

- RolePrincipal using binary, 707
- session state and, 441–444

### ServerManager **.NET class, 7**

- serverSearchTimeout **attribute,**
  - ActiveDirectoryMembershipProvider, **648–649**

### session data partitioning, **420–421**

### session identifiers, logon

- enforcing single logons, 359, 363, 367
- session state vs. logon session, 418

### session IDs

- cookie-based, 421–424
- DoS attacks and, 437–439
- regenerating, 424–426
- reuse and expired sessions, 435–436

### session state, **417–448**

- configuring inside IIS 7.0, 426–427

- cookie-based session IDs, 421–424

- cookieless sessions, 424–426

- database security for SQL Server, 445–446

- Facade pattern, 483

- Integrated mode and, 427–435

- logon sessions vs., 417–420

- protection from DoS attacks, 437–439

- security for OOP state server, 447

- session data partitioning, 420–421

- session ID reuse and expired sessions, 435–436

- Strategy pattern, 474

- summary review, 447–448

- trust level restrictions for, 439–444

### **session-based cookies, 288**

### `<sessionstate>` **configuration section, 426–427**

- set\_defaultLoadCompletedCallback( )  
**function, 816**

- set\_defaultLoginCompletedCallback()  
**function, 807**

- set\_defaultLogoutCompletedCallback()  
**function, 807**

- SetAuthCookie, FormsAuthentication

- AutoDetect cookieless option, 312

- cookie-based SSO-lite, 352, 354

- HttpOnly cookies, 306

- Set-Cookie **command, 303–305**

- SetDirty **method**

- RolePrincipal class, 698, 705

- SqlRoleProvider, 754

- SetReadOnly, ProviderCollection, **488**

### **SHA1 algorithm**

- encryption in ASP.NET 2.0/3.5, 300

- security of signed tickets, 295–299

- SqlMembershipProvider, 588

### **SIDs (security identifiers),**

- WindowsTokenRoleProvider, **727–728**

### **signed tickets, 295–299**

### **signing, precompiled assemblies, 455–457**

### **Singleton pattern, provider model, 481–482**

### **site navigation, 462–467**

- Facade pattern, 483

- security, 462–467

- Strategy pattern, 474

- SiteMapNode, **site navigation, 462–467**

- SiteMapProvider(s), **site navigation, 462–463**

- SkipAuthorization **property**

- FormsAuthenticationModule, 116–117

- UrlAuthorizationModule, 125

- Sleep **class, 102–104, 107–108**

### **sliding expirations, forms authentication**

- cookie-based SSO-lite, 346, 351, 357

- EndRequest, RoleManagerModule, 711

- never using with cookieless tickets, 316

- overview of, 308

- `SmtpPermission` **class**, 193
- `SocketPermission` **class**, 193–194
- Specific user, ASP.NET impersonation**, 97
- spoofing threats**, 825
- SQL injection attacks**, 849–853
- SQL Server**
  - database security, 445–446, 843–845
  - session ID DoS threats, 437–438
  - `SqlMembershipProvider` configuration, 576–577
  - `SqlRoleProvider` configuration, 737–738
  - using database-based policy store, 780–783
- SQL Server Express. See SSE (SQL Server Express)**
- `SqlClientMembershipProvider`, 550–551, 552–553
- `SqlClientPermission`
  - sandboxed access to ADODB, 209
  - sandboxed access to `SqlClient`, 212–214
  - serialization, 443–444
  - `SqlRoleProvider` in Low trust, 739
  - `SqlRoleProvider` in partially trusted non-ASP.NET, 742, 745
  - working with trust levels, 164–165
- SQLEXPRESS**, 577
- `SqlMembershipProvider`, 561–637
  - account lockouts, 617–621
  - changing password formats, 588–590
  - common database schema, 562–573
  - configuring ASP.NET membership, 792–793
  - custom password generation, 590–594
  - database schemas and dbo user, 586–588
  - database security, 584–586
  - date-time values in, 536–537
  - enforcing password history, 602–617
  - enforcing password strength, 598–600
  - hooking `ValidatingPassword` event, 600–601
  - implementing automatic unlocking, 621–626
  - implementing custom encryption, 594–597
  - managing an application's users through IIS 7.0, 632–636
  - Membership database schema, 573–577
  - of `MembershipProvider` base class, 537
  - overview of, 561
  - SQL Server configuration, 576–577
  - summary review, 637
  - supporting dynamic applications, 626–632
  - working with SQL Server Express, 577–584
- `SqlRoleProvider`, 735–760
  - ASP.NET role management, 795
  - authorizing with roles in data layer, 755–757
  - database schema, 735–739
  - database security, 745
  - with limited set of roles, 748–755
  - managing roles through IIS 7, 758–760
  - overview of, 735
  - summary review, 760–761
  - supporting dynamic applications, 757–758
  - trust-level checks, 739–745
  - in Windows authentication, 746–748
- SSE (SQL Server Express)**, 577–584
  - connection string, 584–585
  - issues with sharing, 582–583
  - overview of, 577–582
- SSL (Secure Socket Layer)**
  - cookieless identifiers and, 425–426
  - encrypting data using, 848
  - `EndRequest`, `RoleManagerModule`, 711
  - implementing automatic unlocking, 621
  - installing ADLDS, 643–644, 675–676
  - installing ADLDS with application partition, 677, 680, 683, 689–690
  - protecting sensitive data exchanges, 871
  - tracking logon status, 419–420
- SSO (single sign on)**
  - cross-application redirects, 334–335
  - cross-application sharing of ticket, 333–334
- SSO-lite, cookie-based**, 342–358
  - central login application, 351–355
  - examples of using, 356–357
  - final leg of login, 355–356
  - overview of, 342–346
  - sample application, 346–351
  - summary review, 357–358
- stack frames**, 159–160, 164
- `StaticFileHandler`, 384
- `StatusCode` **property**, `DefaultAuthentication`, 118–119
- STIDE acronym**, 825–826
- stored procedures, password history**, 603–604
- storing application name, database schema**, 562–563
- Strategy pattern, provider model**, 472–474
- string replacements, policy file**, 156–157
- strongly named assemblies**
  - APTCA and, 200–202
  - APTCA and bin directory, 202–204
  - sandboxing with, 204–208
- strongly typed configuration API**, 244, 254–255
- surf (cross-site request forgery)**, 857–861
- synchronization, of key containers across machines**, 272–274
- `Sys.Service.RoleService` **class**, 817
- `Sys.Services._AuthenticationService` **class**, 802, 804–807
- `Sys.Services._RoleService` **class**, 802–803
- `Sys.Services.RoleService` **class**, 815–816
- `<system.applicationHost />` **section group**
  - `ApplicationHost.config` file, 3–4
  - IIS 7.0 configuration, 234
  - IIS 7.0 feature delegation, 238–243
- System.Configuration classes**, 490–494
- System.Configuration.Provider classes**, 484–489

# System.Web.ApplicationServices.AuthenticationService class

---

System.Web.ApplicationServices  
  AuthenticationService **class**, 804, 805  
System.Web.ApplicationServices.  
  AuthenticationServices.RoleService  
  **class**, 815  
**System.Web.Configuration classes**, 489  
<system.web.extensions /> **configuration**  
  **section**, 799–800  
System.Web.IisTraceListner **class**, 16–17  
<system.webServer /> **configuration**  
  **group**  
  <globalModules>, 34–35  
  <modules>, 35–37  
  ApplicationHost.config file, 4–6  
  authenticating classic ASP, 395  
  configuring AnonymousAuthenticationModule,  
    85–87  
  defined, 34  
  enabling ASP.NET applications with AJAX, 800–801  
  feature delegation, 238–243  
  IIS 7.0 configuration, 234–235  
  managed modules/handlers, 67, 77, 236

## T

**tampering threats**, 825

**tempdb**, 445–446

**thread identity**

- asynchronous pipeline events and, 100–110
- establishing for OS, 92–98
- forms authentication and OS, 117
- locating for requests, 87–92

**threats**. *See* **ASP.NET security, web application best practices**

**3DES encryption**. *See also*

- RsaProtectedConfigurationProvider
- ASP.NET 1.0 and 1.1, 299
- ASP.NET 2.0 and 3.5, 300
- sharing tickets between ASP.NET versions, 324–325

**tickets**

- cookieless. *See* **cookieless forms authentication**
- passing across applications. *See* **forms authentication, passing tickets across applications**
- persistent, 288–291
- security of signed, 295–299
- sharing between versions, 324–325

**tilde syntax, file-based policy stores**, 768

**timeouts**

- commandTimeout, SqlRoleProvider, 737
- cookie-based SSO-lite, 351
- cookieless tickets, 317
- enforcing forms authentication expiration, 291–294
- setting cookie, 289–290

ToEncryptedTicket **method**

- RoleManagerModule, 711
- RolePrincipal class, 702–703, 707

**tracing system**, 12–17

**tracking online users**, MembershipProvider,  
549–550

**transaction behavior**, SqlRoleProvider, 738–739

TransactionScope **class**, ADO.NET 2.0, 617

**transmission, security of data**, 871

**troubleshooting**

- complex permission sets, 165–167
- with Failed Request Tracing feature, 12–16
- with Runtime Status and Control API, 17
- with System.Web.IisTraceListner class, 17

<trust /> **element**

- choosing permissions using, 221
- configuring trust levels, 150–151, 221–222
- finding trust policy file, 155–156
- processRequestInApplicationTrust, 150–151
- trust levels and session state, 439–441
- using WebPermission with, 176–177

**trust levels**, 147–221

- in action, 162–165
- AllowPartiallyTrustedCallersAttribute,  
198–204

ASP.NET functionality and, 184–185

configuring, 150–151

cookie-based SSO-lite, 358

defining ASP.NET, 148–150

defining CAS, 162

DnsPermission class, 188

EnvironmentPermission class, 188

FileIOPermission, 189

finding policy files, 155–156

intent of, 183–184

IsolatedStorageFilePermission class, 190

LinkDemand exception behavior, 195–198

LINQ, 179–181

overview of, 147

partial trust. *See* **partial trust**

permission sets, 157–158

permission sets, matching to code, 159–161

permissions, default. *See* **permissions, default security**

permissions, individual, 158–159

permissions, troubleshooting complex, 165–167

PrintingPermission class, 190

processRequestInApplicationTrust, 214–221

ReflectionPermission class, 191

RegistryPermission class, 191

sandboxing with strongly named assemblies, 204–214

SecurityPermission class, 192–193

session state and, 439–444

SmtpPermission class, 193

SocketPermission class, 194

SqlClientPermission class, 194

SqlRoleProvider, 739–745  
 string replacements in policy files, 156–157  
 WebPermission class, 194–195  
 working with different, 151–155

**trust levels, customizing, 167–179**

OdbcPermission, 173–176  
 OleDbPermission, 171–173  
 overview of, 167–171  
 using WebPermission, 176–179

**Try/Catch blocks, 863–864**

type **attribute, 35**  
 TypeInitializationException, **504**

**U**

**UI (user interface), 6, 360**

**UNC (Universal Naming Convention) shares**

AspNetHostingPermission outside of ASP.NET, 185  
 configuring ASP.NET impersonation, 97–98  
 FileAuthorizationModule, 123  
 security choices, 81  
 WindowsAuthenticationModule, 114–115

**unified processing pipeline**

AuthenticateRequest, 110–117  
 AuthorizeRequest, 122–135  
 blocking requests at IIS level, 135–137  
 DefaultAuthentication and Thread.  
     CurrentPrincipal, 117–120  
 EndRequest, 143–144  
 IIS 7.0 running in Classic mode, 31–32  
 Integrated mode and, 30, 32–33  
 Integrated mode, per-request security, 81–87  
 PostAuthenticateRequest, 120–122  
 PostAuthorizeRequest, 135  
 synchronous events and stages in, 99–100  
 thread identity and asynchronous pipeline events, 100–110

**Universal Coordinate Time. See UTC (Universal Coordinate Time)**

**Universal Naming Convention. See UNC (Universal Naming Convention) shares**

**unlocking, automatic,, 621–626**

UnlockUser **method**  
 ActiveDirectoryMembershipProvider, 653  
 MembershipProvider, 546  
 SqlMembershipProvider, 620, 623–624

**unmanaged (native) modules, 22–25**

UpdateCache **method, AzMan, 766**

**updates**

of LastActivityDate column, 565–566  
 Membership state after, 529–534  
 why only certain properties are updatable, 534–535

**UpdateUser **method****

ActiveDirectoryMembershipProvider, 653

tracking online users, 550  
 user creation and user, 541–543

**UPNs, 659–660, 786**

UrlAuthorizationModule  
 configuring, 40–42  
 forms authentication tasks, 288  
 managed, 124–129  
 managed vs. native, 134–135  
 managed vs. native configuration, 237–238  
 native, 129–135  
 overview of, 11  
 using forms authentication across different content types, 328

**URLs**

cookie-based SSO-lite and. See SSO-lite, cookie-based  
 cookieless forms authentication for classic ASP, 391–392  
 cookieless tickets and, 315–321  
 session ID reuse and expired sessions, 435–436

**URLScan security add-on, 11, 135**

UseCookies, cookieless **attribute, 309**

UseDeviceProfile

cookieless attribute, 309  
 cookieless forms authentication, 313–315  
 issuing cookieless session IDs, 424

useMachineContainer, **RSA provider, 268, 271–272**

useMachineProtection, **DPAPI provider, 265–267**

useOAEP, **RSA provider, 268**

**user input validation, best practices, 829–838**

ASP.NET validation controls, 829–831  
 encoding and filtering, 831–834  
 protecting against cross-site scripting, 854  
 protecting against SQL injection, 850–851  
 request validation, 835  
 verifying data input, 835–838

**user instances, SSE, 577–581**

**user interface (UI), 6, 360**

**User level, CAS policies for, 162**

UserData **property, 329–332**

UserId **column, 574, 737**

UserIsOnlineTimeWindow **property, Membership class, 521**

UserIsOnlineTimeWindow **property, MembershipProvider, 549**

UserName **property, MembershipUser**

AD and ADLDS directory schema mappings, 646–647  
 defined, 524  
 updatability of, 524, 534

**usernames**

AD and ADLDS connection settings, 645  
 affecting URL authorization, 129  
 database security, 585  
 passing to ASP, 394  
 payload size with cookieless tickets and, 320  
 primary key for, 552–553

## userPrincipalName attribute

---

userPrincipalName **attribute, AD LDS, 689**

### users

- authentication best practices, 827–828
- authorization best practices, 828
- common table for, 563–566
- configuring in native
  - UrlAuthorizationModule, 131
- creating and deleting AD and AD LDS, 641–642
- cross-application sharing of tickets, 333–334
- database schemas and dbo, 586–588
- linking custom features to records of, 569–572
- managing with .NET Users applet, 632–636
- passing roles to classic ASP, 397–398

**users, MembershipProvider class**

- creating and updating, 541
- managing SqlMembershipProvider, 632–636
- retrieving and searching for multiple, 544–545
- retrieving data for single, 544
- supporting self-service password reset or retrieval, 547–549
- tracking online, 549–550
- validating credentials, 545–547

UseUri, cookieless **attribute, 309**

**UTC (Universal Coordinate Time)**

- account lockouts, 617–620
- enforcing in Membership, 536–537
- ticket expiration, 291–292

## V

Validate **method, Page, 831**

ValidateRequest **property, Page class, 835, 854–857**

ValidateUser **method**

- ActiveDirectoryMembershipProvider, 653, 673
- AuthenticationService class, 805
- MembershipProvider, 546, 549
- SqlMembershipProvider, 624–626

ValidatingPassword **event**

- MembershipProvider, 542–544
- SqlMembershipProvider, 598, 600–601

**validation. See also user input validation, best practices**

- credentials, MembershipProvider class, 545–547
- fraudulent postbacks, 460–462
- ProviderCollection read-only, 488
- request, 450–451
- RolePrincipal class, 704–705

validation **attribute, viewstate protection, 451–453**

validationkey **attribute**

- encryption in ASP.NET 2.0 and 3.5, 300
- overview of, 295–296
- sharing tickets between versions, 324

ValOnValidatingPassword **method, MembershipProvider, 544**

### VB.NET

- ActiveDirectoryMembershipProvider, 658–659
- ActiveDirectoryMembershipProvider in partial trust, 685–687
- ActiveDirectoryMembershipUser, 656
- AJAX custom authentication service, 813–814
- AJAX custom role service, 819
- AllowPartiallyTrustedCallersAttribute, 199, 201, 203
- ASP.NET membership, 794
- ASP.NET role management, 795
- AspNetHostingPermission class, 185–187
- asynchronous page tasks, 143
- asynchronous pipeline events, 102–106, 108
- asynchronous PreRender processing, 138–141
- authenticating classic ASP with ASP.NET, 390
- authorizing classic ASP with ASP.NET, 397–398
- building provider-based feature, 496–503, 506–509, 511–512, 515–516
- clock resets and, 293
- container nesting, 661–662
- cookie settings, 304–305
- cookie timeout, 290
- cookie-based SSO-lite, 347–350, 352–354
- cooked cross-application behavior, 341–342
- cookieless cross-application behavior, 335, 337–338
- cookieless forms authentication for classic ASP, 391
- cross-site request forgery, 858–860
- cross-site scripting threat in AJAX, 876–877
- custom Hash algorithms, 558–559
- custom password encryption, 594–597
- custom passwords, 592–594
- customizing configuration providers, 279, 282–285
- customizing OdbcPermission, 174–175
- customizing OleDbPermission, 172–173
- directory-based policy store, 776, 778
- encrypting data, 847–848
- enforcing logouts, 369–371
- enforcing single logins, 362–367
- Factory Method pattern, 477–479
- filtering data before encoding, 833–834
- fraudulent postbacks, 459
- generating keys, 302–303
- global error handling, 864–865
- hash Helper, 408–410
- IIS 7 wildcard mappings, 379, 381
- JSON hijacking threat, 874–875
- LinkDemand exception behavior, 195–197
- LINQ trust levels, 180
- local configuration, 251
- locating identity for requests, 88, 90–91
- managed handlers, 50–51, 55–56, 58–63, 65–66, 71–75

- managed modules, 68
  - Membership supported environments, 556
  - MembershipProvider class, 539–540
  - Microsoft.Web.Administration, 7–8
  - migrating ASP.NET, 44, 47–48
  - partial trust, 253
  - passing data to ASP from ASP.NET, 392–393
  - passing data to classic ASP, 399–406, 412–413
  - permission sets, 165–167
  - persistent cookies, 289
  - policy file permissions, 168–171
  - PostAuthenticateRequest, 708, 710
  - preventing SQL injection, 850–853
  - processRequestInApplicationTrust, 216, 219–220
  - protected configuration in partial trust, 276–278
  - protecting against cross-site scripting, 855–857
  - reading and writing configuration, 244–247
  - Role Manager with membership, 786–789
  - RoleManagerModule, 719–721
  - RolePrincipal class, 696, 701, 703–704
  - RoleProvider class, 722–723
  - Roles.DeleteCookie, 693
  - sandboxed access to ADODB, 209–211
  - secure cookies, 839–841
  - securing containers, 667
  - self-service password resets, 673
  - serialization and, 442–444
  - session state in Integrated mode, 430–432
  - signed tickets, 297–298
  - simulating DOS attack, 866–871
  - site navigation security, 467
  - SqlMembershipProvider, automatic unlocking, 622–625
  - SqlMembershipProvider, dynamic applications, 627, 629–630
  - SqlMembershipProvider, password history, 606–617
  - SqlMembershipProvider, password strength, 599–601
  - SqlRoleProvider, dynamic applications, 757–758
  - SqlRoleProvider, limited set of roles, 748–754
  - SqlRoleProvider in partially trust non-ASP.NET, 740–741
  - SqlRoleProvider in Windows authentication, 746–748
  - System.Configuration classes, 490, 493–494
  - System.Configuration.Provider classes, 484, 486–487, 489
  - System.Web.Configuration classes, 489
  - tracing system, 15–16
  - trust levels, 152–154
  - Try/Catch blocks, 864
  - UserData property, 330–332
  - validating user input, 831
  - verifying data input, 836–838
  - WebPermission, 177–179
  - WindowsTokenRoleProvider, 728, 730–732
  - verifying data input, 836**
  - Version property, RolePrincipal class, 698
  - versioning provider schemas, 566–568**
  - views**
    - Membership database schema, 576
    - querying common tables with, 568
  - viewstate protection, 451–454**
  - vw\_aspnet\_Applications view, 568
  - vw\_aspnet\_MembershipUsers view, 576
  - vw\_aspnet\_Users view, 568
- ## W
- WAS (Windows Process Activation Service), 20, 27**
  - WAT (Web Administration Tool), 582–583, 754–755, 769**
  - Web Administration Tool (WAT), 582–583, 754–755, 769**
  - Web Parts Personalization**
    - Facade pattern, 483
    - Strategy pattern, 473–474
    - updating LastActivityDate, 565–566
  - Web Services Description Language (WSDL), 871**
  - web.config file
    - configuring
      - ActiveDirectoryMembershipProvider, 657–659
    - configuring trust levels, 150–151
    - installing managed handler, 67, 77
  - WebConfigurationManager class
    - reading and writing configuration, 245–247
    - reading local configuration, 247–249
    - using protected configuration providers in partial trust, 276
    - writing local configuration, 250–251
  - WebPermission, **176–179**
  - WebPermission class, **194**
  - whitelisting, 833–834**
  - wildcard mappings, classic ASP with ASP.NET, 396**
  - wildcard mappings, IIS 7, 375–383**
    - configuring, 376–382
    - overview of, 375–376
    - resourceType setting, 382–383
  - Windows authentication. See also**
    - WindowsAuthenticationModule
    - best practices, 827
    - database security, 446, 584, 842–843
    - IIS Manager feature and, 12
    - reading local configuration, 247–248
    - Role Manager integration with, 553, 565, 697, 709, 794–796

### **Windows authentication (*continued*)**

- site navigation security, 464
- SqlRoleProvider and, 746–748
- using AuthorizationStoreRoleProvider, 765
- using FileAuthorizationModule, 123
- using native AnonymousAuthenticationModule, 86, 112–114, 120
- using SqlRoleProvider, 746–748
- using UrlAuthorizationModule, 124
- using WindowsTokenRoleProvider, 726–732
- writing local configuration, 250–251

**Windows Process Activation Service (WAS), 20, 27**

WindowsAuthenticationModule

- authenticating requests, 111–115
- impersonation token for, 93
- security configuration, 83–84

WindowsIdentity

- enabling Role Manager, 697
- FileAuthorizationModule requiring, 123–124
- WindowsAuthenticationModule, 113–115

WindowsPrincipal

- locating security identity for requests, 91–92
- security choices, 81
- security configuration, 84
- WindowsAuthenticationModule, 111–112
- WindowsTokenRoleProvider, 727

WindowsTokenRoleProvider, **726–732, 733**

**worker processes, per-request security, 82**

### **writing configuration, 244–253**

- overview of, 244–247
- permissions for local, 249–251
- permissions for remote editing, 251–253

**WSDL (Web Services Description Language), 871**

**WSS\_Minimal, 149**

**WWW publishing service, 19–20**

## X

### **XML files**

- IIS 7.0 configuration based on, 3, 233
- using file-based policy store, 768–771
- working in partial trust, 784

XmlSiteMapProvider, **462–463**

### **XSS (cross-site scripting threat)**

- AJAX-enabled, 875–877
- amplified in AJAX-enabled applications, 875–877
- example of, 824–825
- guarding against, 853–857
- overview of, 853

## Z

ZqlClientPermission **class, 194**